25

Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.111 – Introductory Digital Systems Laboratory

Quiz – October 30, 2003

7:30 PM – 9:30 PM Walker, 50-340

Student Name: DAN PORTS

TA (circle one): Matthew Cohen
Neira Hajro
Frank Honore

Problem 1 (40 points): 42 FH

Problem 2 (40 points): 36 DOD

Problem 3 (40 points): 39 DOD

Problem 4 (40 points): 40 FH

Problem 5 (40 points): 38 DOD

Total 195

Problem 1 (40 points) Combinational Logic

a) (16 points)
Provide the MSP and the MPS for the Karnaugh maps below. State whether each is unique or not. Two copies of the K-map are shown for your convenience.

CD \ AB	00	01	11	10
00	1	–	1	–
01	–	–	1	–
11	0	0	0	0
10	–	0	0	1

CD \ AB	00	01	11	10
00	1	–	1	–
01	–	–	1	–
11	0	0	0	0
10	–	0	0	1

MSP = $\bar{C} + \bar{B}\bar{D}$ ✓

MPS = $(C\bar{C} + D\bar{D})(\bar{B} + \bar{C})$ ✓

$\overline{MPS} = CD + BC$

$MPS = \overline{CD + BC}$

$= \overline{CD} \cdot \overline{BC}$

$= (\bar{C} + \bar{D})(\bar{B} + \bar{C})$

Unique

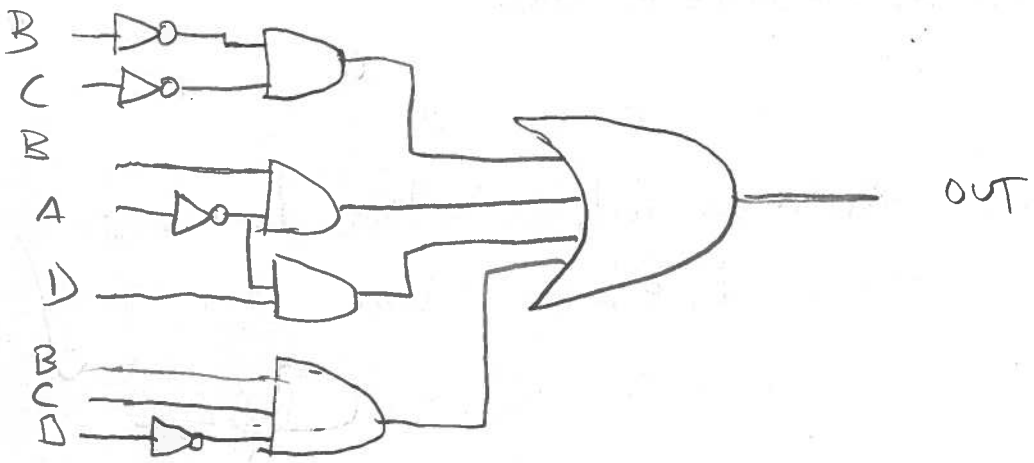
Unique

Problem 1 (continued)

b) (8 points)
Provide a hazard free realization of the combinational logic specified by the Karnaugh map below if only one input changes at a time. Use AND gates and OR gates with any number of inputs.

CD \ AB	00	01	11	10
00	1	–	0	1
01	1	1	0	–
11	1	1	0	0
10	0	1	1	0

$F = \bar{B}\bar{C} + \bar{A}B + \bar{A}D + BC\bar{D}$ ✓



bonus +2

XY	00	01	11	10
Z	0	0	0	1
	1	1	1	0

Problem 1 (continued)

c) (8 points)
Simplify the algebraic expression

$(X \bullet Y) + (Y \bullet Z) + \overline{(X \bullet Z)} =$

$X \cdot Y + \overline{X} \cdot Z$ ✓

d) (8 points)
Under what conditions does the RCO of the 74LS163 have a glitch?
VHDL code for a 74LS163 is on the next page.

When T is high and all Q bits are momentarily high due to a static hazard; i.e. when the count goes from 0111 → 1000 ✓
It may actually be 0111 → 1111 → 1000 if the MSB register changes state. Then there will be a glitch on RCO.

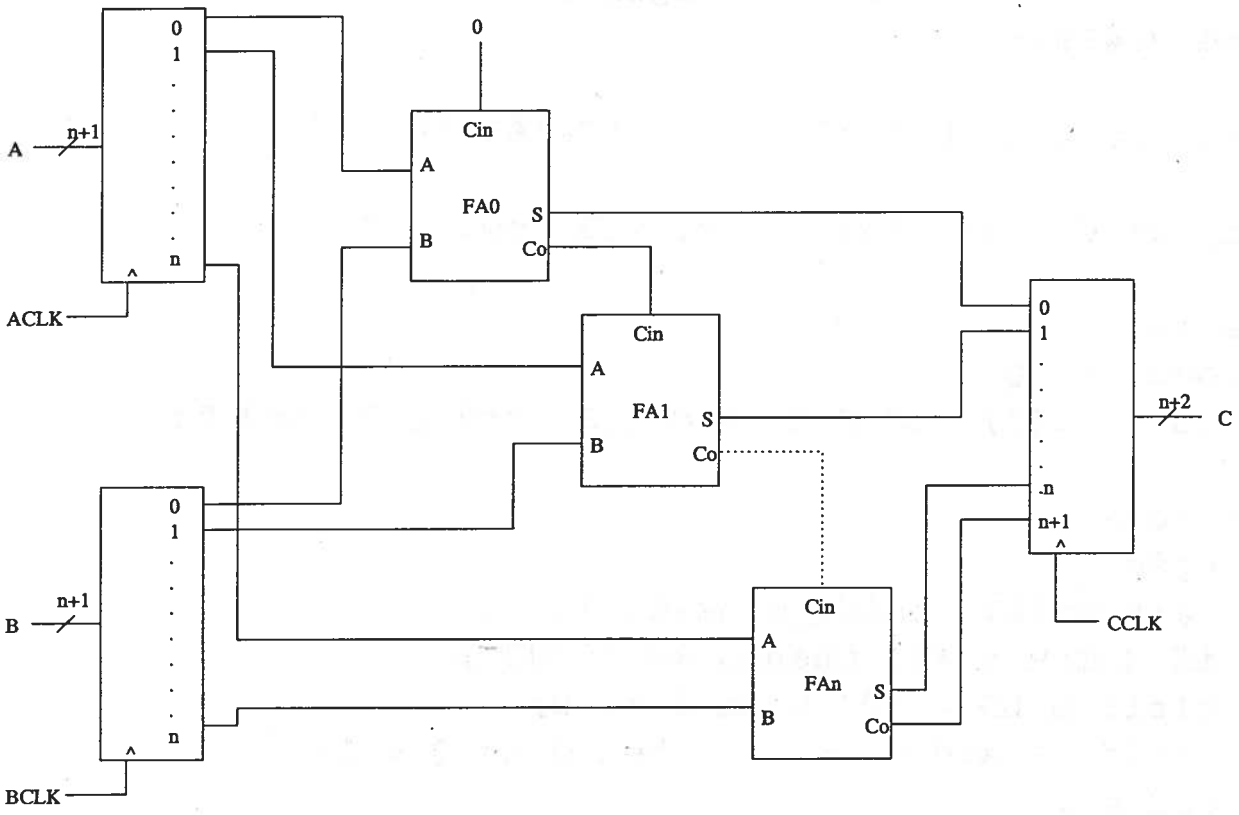
Problem 1 (continued)

For your convenience, here is VHDL code for a 74LS163.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity c74163 is
 port    (N_LD, N_CLR, P, T, CLK: in std_logic;
          D   : in std_logic_vector(3 downto 0);
          count : out std_logic_vector(3 downto 0);
          RCO : out std_logic);
end c74163;

architecture behavioral of c74163 is

signal Q : std_logic_vector(3 downto 0);

begin
 count <= Q;
 RCO <= Q(3) and Q(2) and Q(1) and Q(0) and T;

 process
 begin
  wait until rising_edge(CLK);
  if N_CLR = '0' then Q <= "0000";
  elsif N_LD = '0' then Q <= D;
  elsif (P and T) = '1' then Q <= Q + 1;
  end if;
end process;
end architecture behavioral;
```

Problem 2 (40 points) Fun With Adders

There are n+1 adders in the following circuit.
Full adder parameters: t_{pd_FA} – propagation delay
t_{cd_FA} –contamination delay
Register parameters: t_{su} - setup time
t_h – hold time
t_{pd_FF} - clock to Q propagation Delay
t_{cd_FF} – contamination delay



Problem 2 (continued)

Assume all three clocks are the same for parts a, b, and c, i.e., ACLK = BCLK = CCLK.

a) (8 points)
Provide an expression for the clock period, T, which is guaranteed to work for this circuit.

$T \geq t_{PDFF} + (n+1)\, t_{PD\,FA} + t_{SU}$ ✓

b) (8 points)
Provide an expression for the Register contamination delay for which this circuit is guaranteed to work.

$t_{CD\,FF} + t_{CD\,FA} \geq t_h$ ✓

c) (8 points)
Assume n = 3 for this part.
Give inputs for A and B that result in the minimum guaranteed frequency, equivalently, the maximum adder delay.

A = 1111 B = 1000 X -4

What condition must be satisfied by any such inputs?

Each C_{out} must depend on its C_{in} ⇒ ✓
$\forall i > 0$ exactly one of A_i, B_i is 1.

For parts d and e, suppose ACLK = BCLK and CCLK = ACLK + S

d) (8 points)
What is now your answer for the clock period as in a) above?

$T + S \geq t_{PDFF} + (n+1)\, t_{PD\,FA} + t_{SU}$ ✓

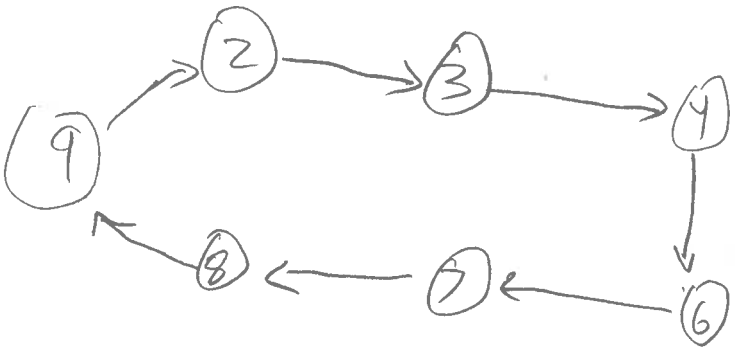
e) (8 points)
What is now your answer for the contamination delay as in b) above?

$t_{CD\,FF} + t_{CD\,FA} \geq t_H + S$ ✓

Problem 3 (40 points) Fun With Counters

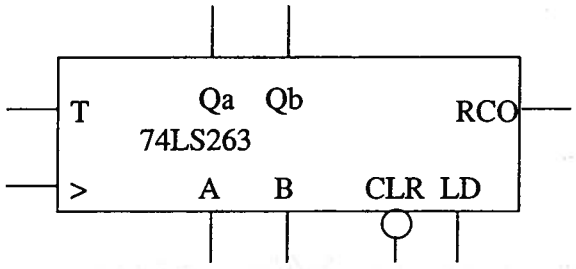
You are hanging out in the 6.111 lab trying to think of a fun way to kill some time. You decide to experiment with counters. Since your telephone number happens to be 234-6789, you decide to create a circuit that counts out your telephone number.

a) (8 points)
Draw a state diagram for your counter. Label each state with a (different) digit of your telephone number.

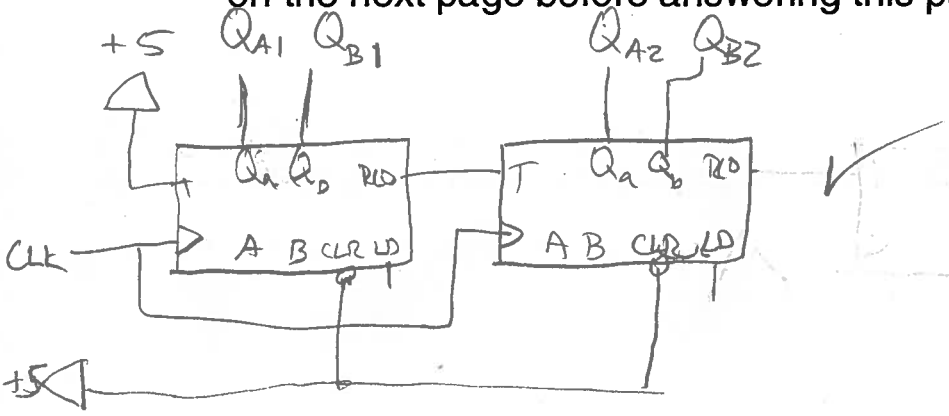


Problem 3 (continued)

You rummage around the lab and find two piles of chips.
One is of 74LS00 which is a quad two-input NAND.
The second pile is of strange chips you have never heard of before, 74LS 263s. You look around for a data sheet but can't find any.
The only thing you can find is a scrap of paper that says
"The 74LS263 is just like the 74LS163 EXCEPT that it is only two-bits wide, the LD input is positive true, and it doesn't have a P input so it counts whenever the T input is asserted high.



b) (8 points)
Wire up enough instances of the 74LS263 to represent the state memory for this counter and the implementation of those states which have a normal counting sequence, i.e., leave the LD input(s) unconnected. (You may want to read the rest of the question on the next page before answering this part.)



The output bits are
Q_{B2} – MSB
Q_{A2}
Q_{B1}
Q_{A1} – LSB

Problem 3 (continued)

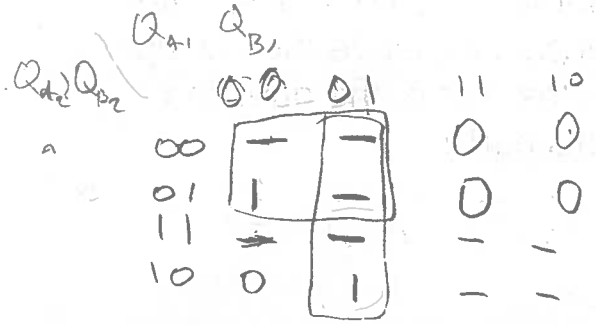
c) (8 points)
What states should the LD inputs be asserted high?

4, 9 ✓

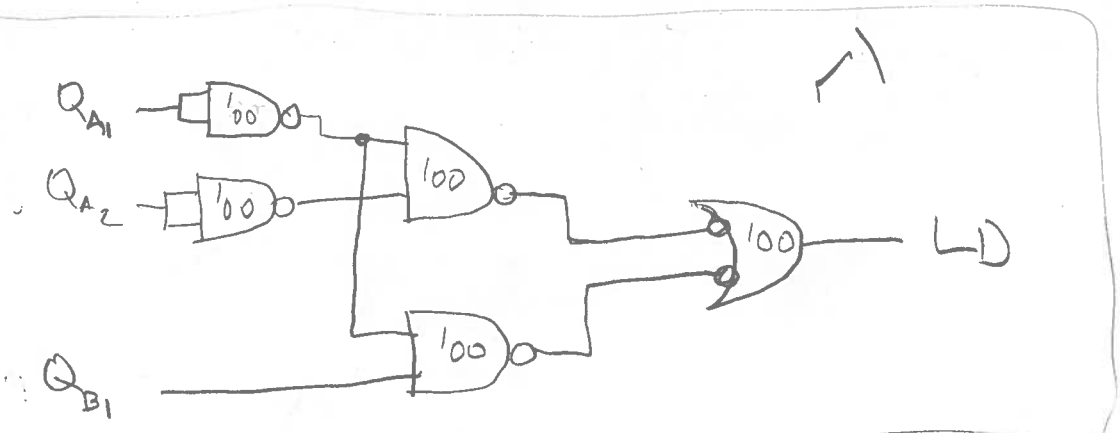
What additional states may the LD inputs be asserted high?

5, 0, 1, 10–15

d) (8 points)
Give a logic diagram with two-input NAND gates for the LD input. Maximum credit is reserved for the simplest (correct) answer.



$$LD = \overline{Q}_{A_1}\,\overline{Q}_{A_2} + \overline{Q}_{A_1}\,Q_{B_1}$$

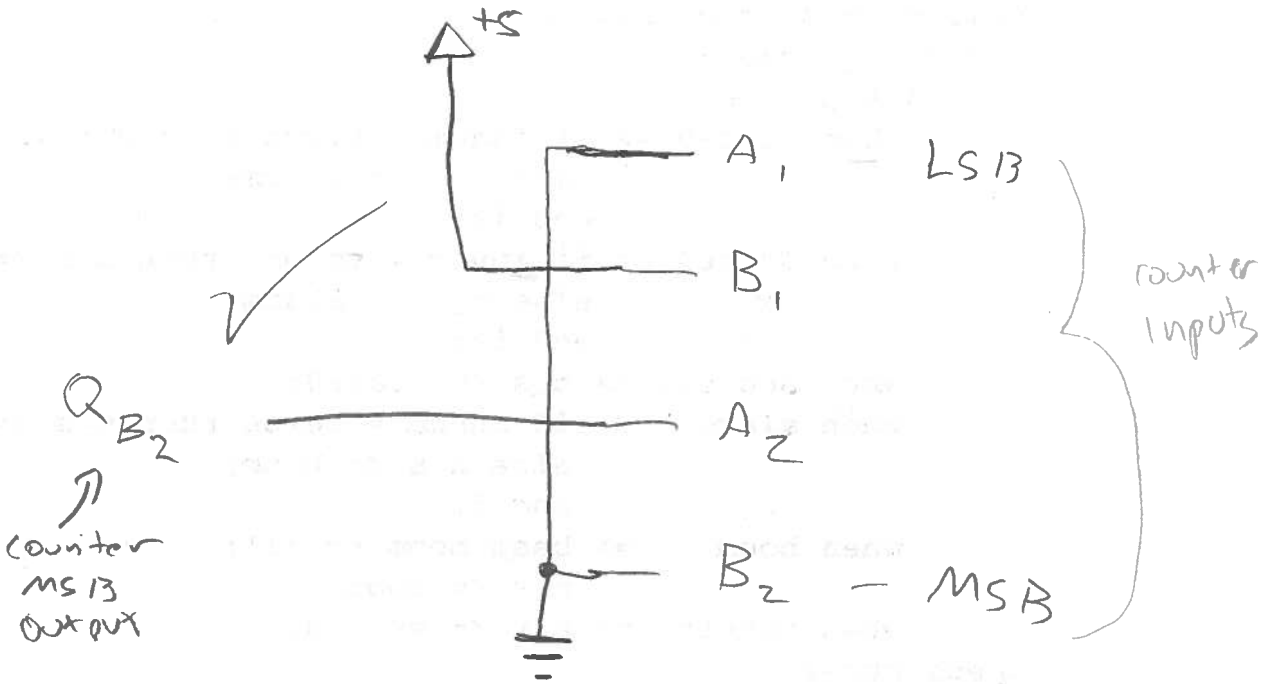


Problem 3 (continued)

e) (8 points)
Give a logic diagram using two-input NAND gates for the four data inputs for the instantiations of the 74LS263 chips you used in part b. Maximum credit is reserved for the simplest (correct) answer.

Should be 6 when count = 4 = 0100 → 0110
2 when count = 9 = 1001 → 0010

Can be anything LD is inactive or in other states since states not used



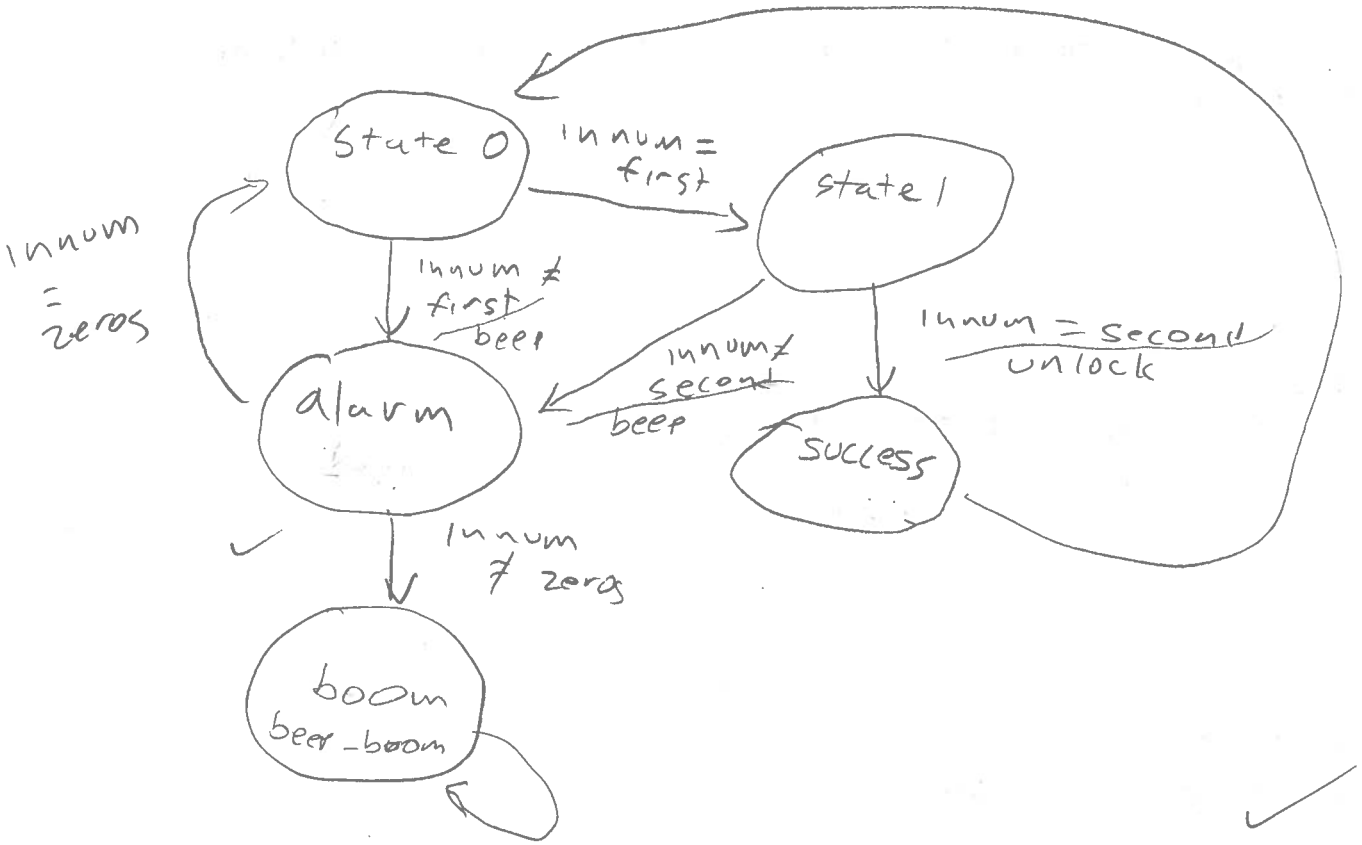
Problem 4 (40 points)

Consider the VHDL program given below.

```
library ieee;
use ieee.std_logic_1164.all;
entity diglock is port(innum      : in std_logic_vector(3 downto 0);
                       enter, clk : in std_logic;
                       beep, beep_boom, unlock : out std_logic);
end diglock;

architecture state_machine of diglock is

signal p_s, n_s  : std_logic_vector(2 downto 0);
signal zeros : std_logic_vector(3 downto 0) := (others => '0');
constant first   : std_logic_vector(3 downto 0):= "0001";
constant second  : std_logic_vector(3 downto 0):= "0101";
constant zeros   : std_logic_vector(3 downto 0):= "0000";
constant state0  : std_logic_vector(2 downto 0):= "000";
constant state1  : std_logic_vector(2 downto 0):= "001";
constant success : std_logic_vector(2 downto 0):= "011";
constant alarm   : std_logic_vector(2 downto 0):= "010";
constant boom    : std_logic_vector(2 downto 0):= "100";

begin
  unlock <= n_s(0) and n_s(1);
  beep <= (not n_s(0)) and n_s(1);
  fsm: process(p_s, innum)
    begin -- case
      case p_s is
        when state0 => if innum = first then n_s <= state1;
                       else n_s <= alarm;
                       end if;
        when state1 => if innum = second then n_s <= success;
                       else n_s <= alarm;
                       end if;
        when success => n_s <= state0;
        when alarm   => if innum = zeros then n_s <= state0;
                        else n_s <= boom;
                        end if;
        when boom    => beep_boom <= '1';
                        n_s <= boom;
        when others  => n_s <= state0;
    end case;
  end process fsm;
  state_clocked: process(clk)
    begin
      if rising_edge(clk) then
       if enter = '1' then p_s <= n_s;
       end if;
     end if;
   end process state_clocked;
end architecture state_machine;
```

Problem 4 (continued)

a) (12 points)
Draw the state diagram for this finite state machine.



Problem 4 (continued)

b) (10 points)
What is another way to specify the initial value of the signal "zeros"?

Could be specified as "0000" instead of (others => 0).

Is the present method better or worse than this alternate method?
Why? Using (others => 0) makes the code somewhat simpler because it doesn't depend on the length of the signal. But the two are equivalent. (Or you could use the constant zeros instead of the signal zeros.)

c) (6 points)
Is beep_boom glitch free? Explain.

Yes. It is set by the fsm process only when p_s = boom.

d) (6 points)
Is beep glitch free? Explain.

No. Can have glitches when the values of n_s(0) and n_s(1) change asynchronously, if one bit changes value before the other.

e) (6 points)
Is unlock glitch free? Explain

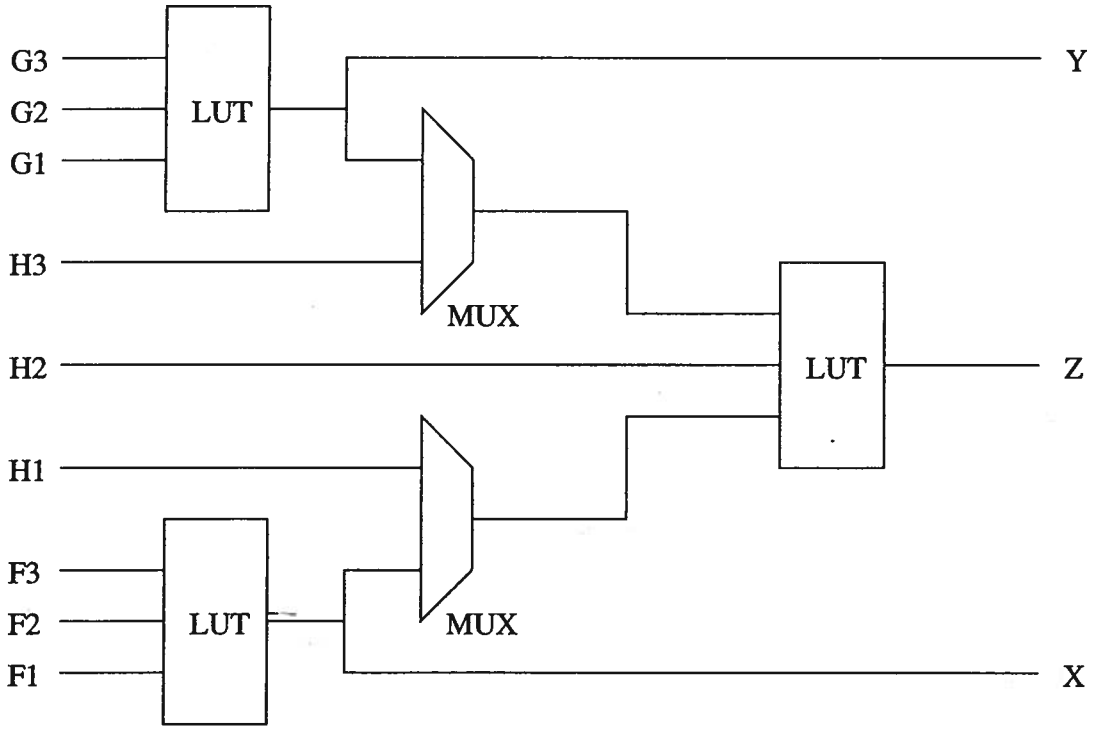
No. Same as beep.

This page is for scratch work (if you need it).

Problem 5 (40 points) Fun with FPGAs

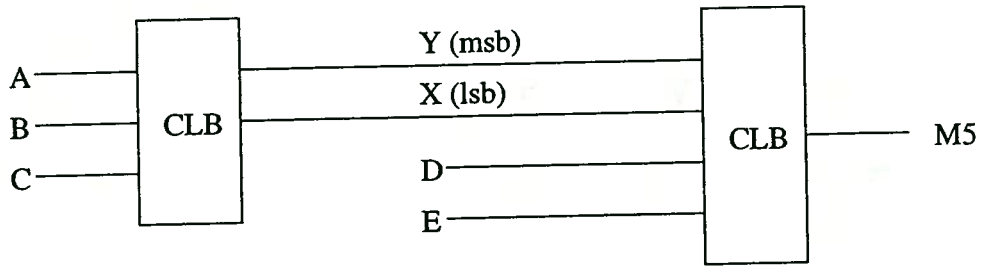
Each Combinational Logic Block (CLB) of a certain FPGA consists of three Look Up Tables (LUTs). Each LUT has eight locations and is addressed by a 3-bit signal. The third LUT is addressable by either outside signals or the output of another LUT. The choice as to which addresses the third LUT is determined by a multiplexor. The select inputs of the multiplexors are configured to 1 or 0 at power up just like the contents of the LUTs.

The architecture of the CLB is given below.

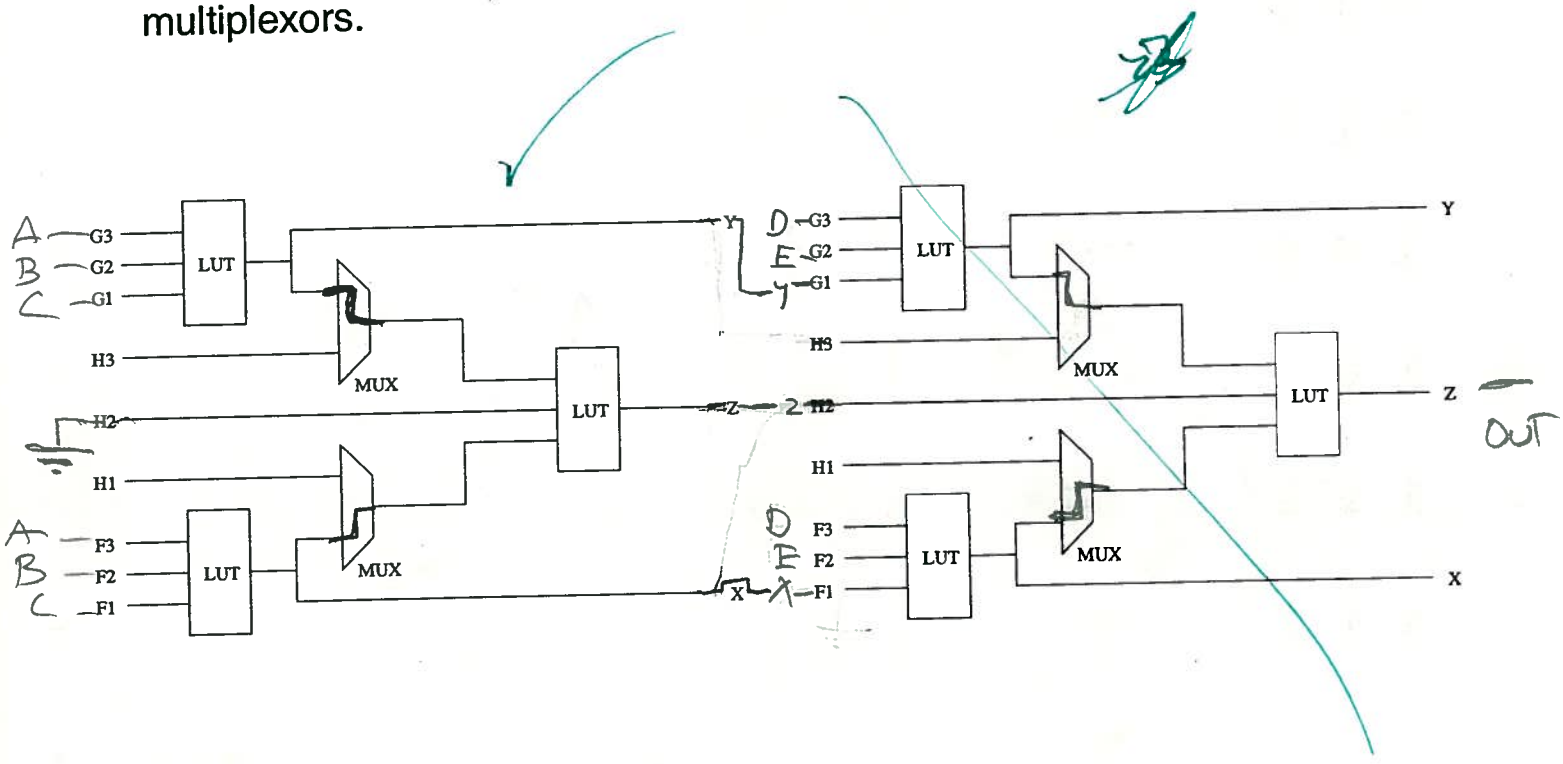


Problem 5 (continued)

A five-input majority circuit can be implemented with two CLBs. The first CLB counts the number of ones from three of the inputs and the second CLB implements a function of these two intermediate signals plus two more of the input signals. A block diagram of these two CLBs is given below.



a) (10 points)
Wire two instances of the CLB on the facing page to implement the above block diagram. Show the signal path through the multiplexors.



Problem 5 (continued)

b) (15 points)
Fill in the three truth tables for the CLB with two outputs. Let X be the least significant bit. Use don't cares where appropriate.

Z = Y X
= H_3 H_1
(H_2 cannot be 1)

A B C (3 2 1)	Z	Y	X
0 0 0	0	0	0
0 0 1	0	0	1
0 1 0	—	0	1
0 1 1	—	1	0
1 0 0	0	0	1
1 0 1	1	1	0
1 1 0	—	1	0
1 1 1	—	1	1

2

Problem 5 (continued)

c) (15 points)
Fill in the three truth tables for the CLB with one output.
Use don't cares where appropriate.

$Y_{out} = Y_{in}(D+E)$
$= G_1(G_2+G_3)$

$X_{out} = X_{in}(DE)$
$= F_3F_2F_1$

$Z_{out} = Y + X + Z_{in}$
$= H_3 + H_2 + H_1$

D	E	X_4	Z	Y	X
3	2	1			
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	1	1

This is the last page of the quiz.